

安装 Babel

由于 JavaScript 社区没有统一的构建工具、框架、平台等等，因此 Babel 正式集成了对所有主流工具的支持。从 Gulp 到 Browserify，从 Ember 到 Meteor，不管你的环境设置如何，Babel 都有正式的集成支持。

本手册的目的主要是介绍 Babel 内建方式的安装，不过你可以访问交互式的[安装页面](#)来查看其它的整合方式。

注意： 本手册将涉及到一些命令行工具如 `node` 和 `npm`。在继续阅读之前请确保你已经熟悉这些工具了。

babel-cli

Babel 的 CLI 是一种在命令行下使用 Babel 编译文件的简单方法。

让我们先全局安装它来学习基础知识。

```
$ npm install --global babel-cli
```

我们可以这样来编译我们的第一个文件：

```
$ babel my-file.js
```

这将把编译后的结果直接输出至终端。使用 `--out-file` 或者 `-o` 可以将结果写入到指定的文件。.

```
$ babel example.js --out-file compiled.js
```

或

```
$ babel example.js -o compiled.js
```

如果我们想要把一个目录整个编译成一个新的目录，可以使用 `--out-dir` 或者 `-d`。.

```
$ babel src --out-dir lib
```

或

```
$ babel src -d lib
```

在项目内运行 Babel CLI

尽管你可以把 Babel CLI 全局安装在你的机器上，但是按项目安装会更好。

有两个主要的原因。

1. 在同一台机器上的不同项目或许会依赖不同版本的 Babel 并允许你有选择的更新。
2. 这意味着你对工作环境没有隐式依赖，这让你的项目有很好的可移植性并且易于安装。

要在（项目）本地安装 Babel CLI 可以运行：

```
$ npm install --save-dev babel-cli
```

注意： 由于全局运行 Babel 是一个坏习惯，如果你要卸载全局安装的版本可以运行：`npm uninstall --global babel-cli`。.

安装完成后，你的 `package.json` 应该如下所示：

```
{
  "name": "my-project",
  "version": "1.0.0",
  "devDependencies": {
    "babel-cli": "^6.0.0"
  }
}
```

现在，我们不直接从命令行运行 Babel 了，取而代之我们将把运行命令写在 **npm scripts** 里，这样可以使用 Babel 的本地版本。

只需将 "scripts" 字段添加到你的 package.json 文件内并且把 babel 命令写成 build 字段。

```
{
  "name": "my-project",
  "version": "1.0.0",
+  "scripts": {
+    "build": "babel src -d lib"
+  },
  "devDependencies": {
    "babel-cli": "^6.0.0"
  }
}
```

现在可以在终端里运行：

```
npm run build
```

这将以与之前同样的方式运行 Babel，但这一次我们使用的是本地副本。

babel-node

如果你要用 node CLI 来运行代码，那么整合 Babel 最简单的方式就是使用 babel-node CLI，它是 nodeCLI 的替代品。

但请注意这种方法并不适合正式产品环境使用。直接部署用此方式编译的代码不是好的做法。在部署之前预先编译会更好。不过用在构建脚本或是其他本地运行的脚本中是非常合适的。

首先确保 babel-cli 已经安装了。

```
$ npm install --save-dev babel-cli
```

注意：如果你不清楚为什么要安装在本地，请阅读上面[在项目内运行 Babel CLI](#)的部分。

然后用 babel-node 来替代 node 运行所有的代码。。

如果用 npm scripts 的话只需要这样做：

```
{
  "scripts": {
-    "script-name": "node script.js"
+    "script-name": "babel-node script.js"
  }
}
```

要不然的话你需要写全 `babel-node` 的路径。

```
- node script.js
+ ./node_modules/.bin/babel-node script.js
```

提示：你可以使用 `npm-run`。

babel-core

如果你需要以编程的方式来调用 Babel 的 API 进行转码，就可以使用 `babel-core` 这个模块。

我们首先来安装 `babel-core`。

```
$ npm install babel-core
var babel = require("babel-core");
```

字符串形式的 JavaScript 代码可以直接使用 `babel.transform` 来编译。

```
babel.transform("code();", options);
// => { code, map, ast }
```

如果是文件的话，可以使用异步 api:

```
babel.transformFile("filename.js", options, function(err, result) {
  result; // => { code, map, ast }
});
```

或者是同步 api:

```
babel.transformFileSync("filename.js", options);
// => { code, map, ast }
```

要是已经有一个 Babel AST（抽象语法树）了就可以直接从 AST 进行转换。

```
babel.transformFromAst(ast, code, options);
// => { code, map, ast }
```

对于上述所有方法，`options` 指的都是 <http://babeljs.io/docs/usage/options/>